

# How to Scale Agile Across Departments with JIRA

WHITEPAPER

 ZPHYR

Once associated only with small application development projects and co-located teams of 8-10 members, the Agile methodology is increasingly being adapted for large-scale enterprise development. Agile is all about getting high-quality software into the hands of your customers as quickly as possible, something that is hard to do on large, enterprise-wide projects that need to scale across a wide variety of departments, locations, lines of business, platforms and technologies. One popular strategy for dealing with scaling issues on agile projects involves releasing a prototype, or minimum viable product (MVP), to get feedback by tracking usage patterns, which is a way to test a product hypothesis with minimal resources right away. Every release going forward can then be measured for how well it converts into the user behaviors you want the release to achieve--which offers great competitive advantage to your organization. The concept of a baseline MVP product that contains just enough features to solve a specific business problem will also reduce wasted engineering hours and a tendency for feature creep or 'gold plating' among large, distributed software teams.

When scaling agile, it's important to use an approach that makes the most sense for your entire IT organization. This may very well involve taking advantage of an appropriate large-scale agile frameworks and/or body of knowledge such as the [Scaled Agile Framework \(SAFe\)](#), the [Disciplined Agile Delivery \(DAD\)](#), or [Large Scale Scrum \(LeSS\)](#). All three of these agile scaling frameworks offer multi-level training and certification, which make them attractive to enterprises that want to expand a small agile group into an enterprise-wide practice and need formal guidance.

Since JIRA is platform and methodology agnostic, it's also possible to take a less formal approach using epics, or large user stories, which can be cross team and involve multiple JIRA projects.

# Table of Contents

- 04** Use Epics as teams
- 05** Categorize departments as projects
- 05** Create custom 'department' field
- 06** Reports and add-ons for Project Management
- 07** Use comments or subtasks for testing notes
- 07** Adopt sprints for marketing tasks

# Use Epics as teams

An epic is really just a large body of work that needs to be done. It's essentially a big user story that can be broken down into a number of smaller stories that, in turn, can be assigned to one or more agile teams. It may take several sprints to complete an epic. Epics are often created for different focus areas of a project such as automation, performance or usability, and all the requirements in the form of user stories are then mapped to that epic focus area. This makes tracking the specific work area of a project easier.

## REAL SCENARIO

This Agile concept can be actively applied across teams. In a sprint, you create multiple epic teams who take care of the user stories or features selected from the epic. For example, one team may work on an automation epic, another on the Performance Improvement epic and another on Usability Improvement epic.

This can be further broken down into Projects or Minimum Viable Products (MVPs). At the end of each MVP, the epic team can add multiple features based on customer demand.

## USE AN EPIC BURNDOWN CHART FOR MEASURING PROGRESS

An epic burndown chart can help you monitor epic teamwork, including things like

- ▶ the progress of each agile team in terms of story points attained during sprints;
- ▶ help estimating how many sprints are required to complete an epic based on epic team velocity;
- ▶ guidance on how more or less resources might impact the epic burndown chart.

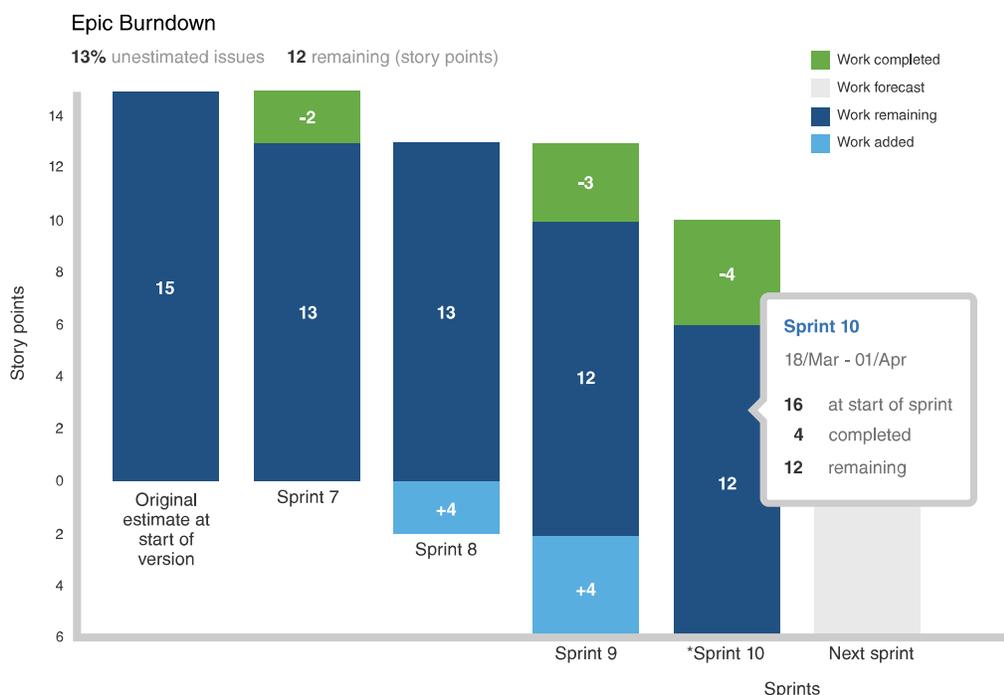


Image Source: <https://www.atlassian.com/agile/delivery-vehicles>

# Categorize departments as projects

A JIRA Project can be defined as a collection of issues. Typical examples of JIRA projects are marketing campaigns, automation projects, software development projects, etc.

## REAL SCENARIO

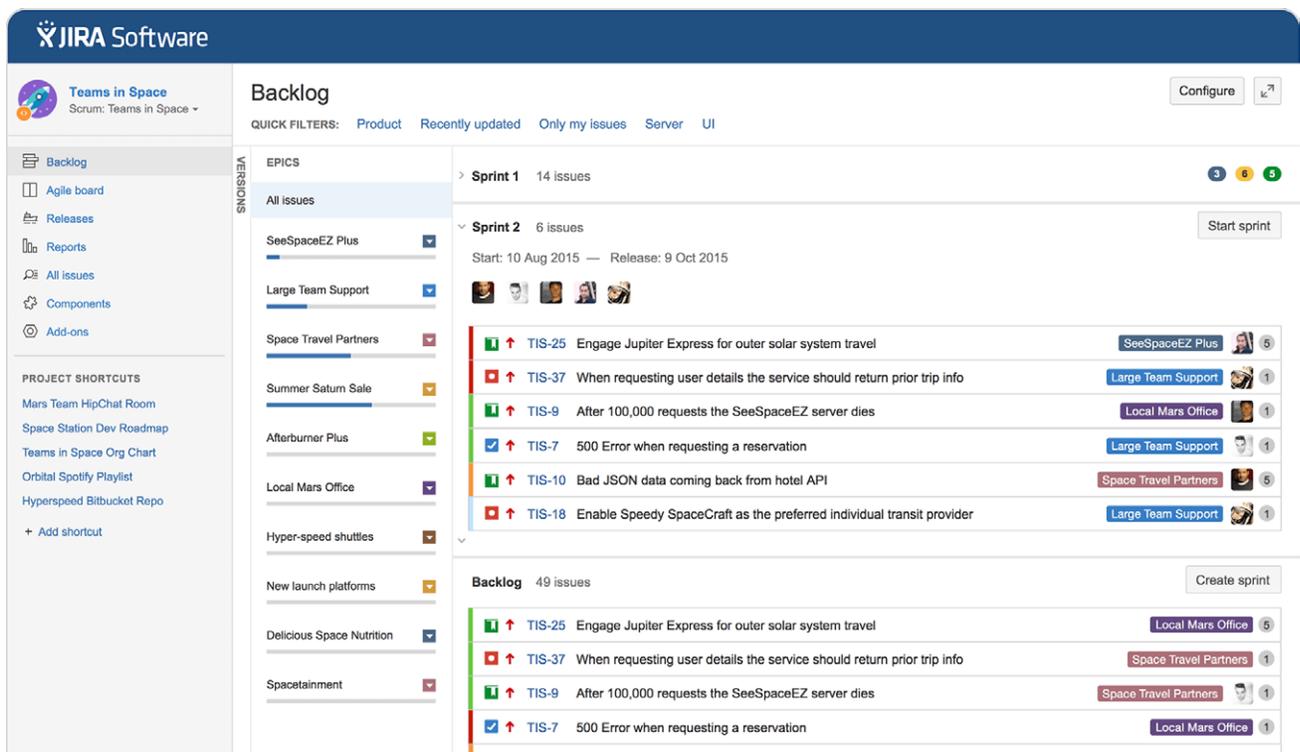


Image Source: <https://www.atlassian.com/software/jira>

## CREATE CUSTOM 'DEPARTMENT' FIELD

One good way to do a scaled agile project across departments is to first define an epic and then use a top-down approach to define roles for the cross-department teams working on the epic. Since JIRA allows you to add custom fields to your application, these can be used to arbitrarily denote departments, such as software development, marketing, etc.

For example, an epic page might be organized with a Product Team that starts by filling in product requirements (use cases, why we need certain features and what are the benefits). Next, a Design Team might chip in usability details, say, pixel size and color. Finally, an Engineering Team could break down into sprints the broad front- and back-end

tasks needs to meet the product requirements.

Using a custom field labeled “Other Considerations” is another way to scale agile across departments. An ‘Other Considerations’ field can help the product owner prioritize tasks on the epic backlog by incorporating input and feedback from customers, designers, and development team(s). A field like this can help the epic product owner determine

customer priority or how difficult different work items are (e.g., is task B is easier if we do A first). This is especially useful in large organizations with multiple Engineering, Operations and Marketing teams where one team might have specific expertise such as familiarity with Big Data Hadoop clusters.

## REPORTS AND ADD-ONS FOR PROJECT MANAGEMENT

JIRA has excellent dashboard gadgets to keep track of projects, such as:

- ▶ **Two-dimensional filer statistics** gadget- Helps in tracking which feature in the software projects is defective/bug free
- ▶ **Road Map** gadget-Provides Summary of progress of the issue/project
- ▶ **Created vs Resolved** Gadget- Shows Difference between issues created and resolved can be tracked
- ▶ **Active Sprint Health** Gadget -Highlights overall health of the sprint

In addition to these versatile gadgets, there are two other valuable options that help in project management

- ▶ the ability to **drill down** to check which issue has failed or which are the most buggy features during software development;
- ▶ the ability to **customize** and edit a gadget. For instance, total defects open in a sprint can be categorized by components versus assignee as well as components versus priority. This flexibility can provide the right information at the right time for efficient project management.

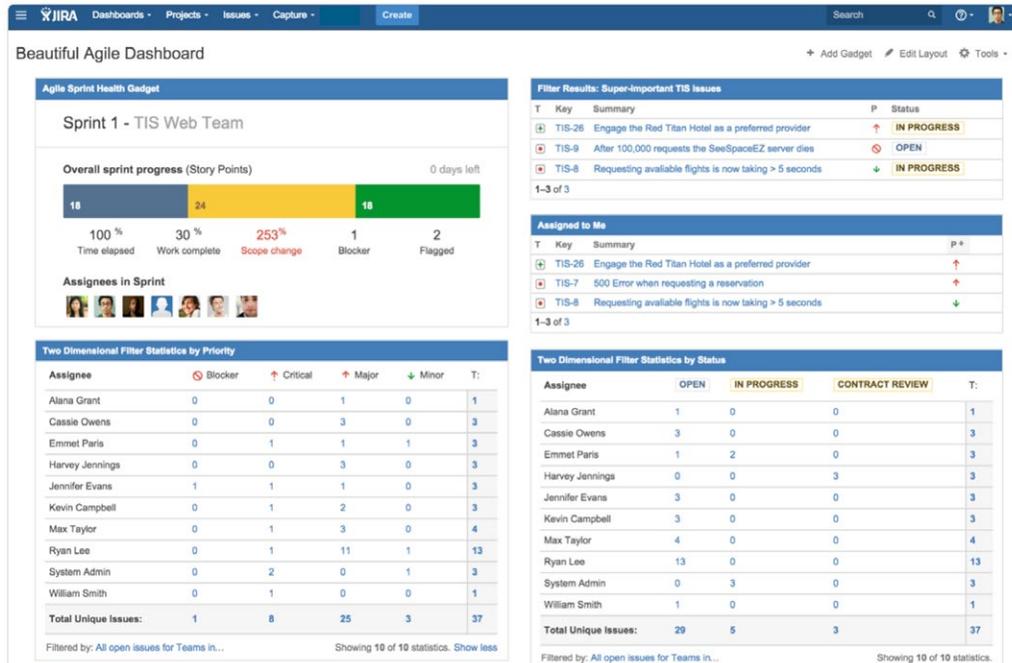


Image Source: <https://www.atlassian.com/blog/jira-software/7-steps-to-a-beautiful-and-useful-agile-dashboard>

## USE COMMENTS OR SUBTASKS FOR TESTING NOTES

A sub-task can be created for an issue in JIRA to either split the issue into smaller chunks, or to allow various aspects of an issue to be assigned to different people. If you're assigning this work to different people in different departments, you can use user comments, a sub-task option in JIRA, to help your teams collaborate.

### REAL SCENARIO

For example, a QA team doing security testing could add user comments to help outline multiple end-user scenarios, each of which can have reports, owners and fields mapped to it. This allows for efficient tracking and management of work being done on the sub-task. For instance, if testers are validating multiple builds to ensure a defect is resolved, comments can be used to communicate the current standing of the issue to the engineering team.

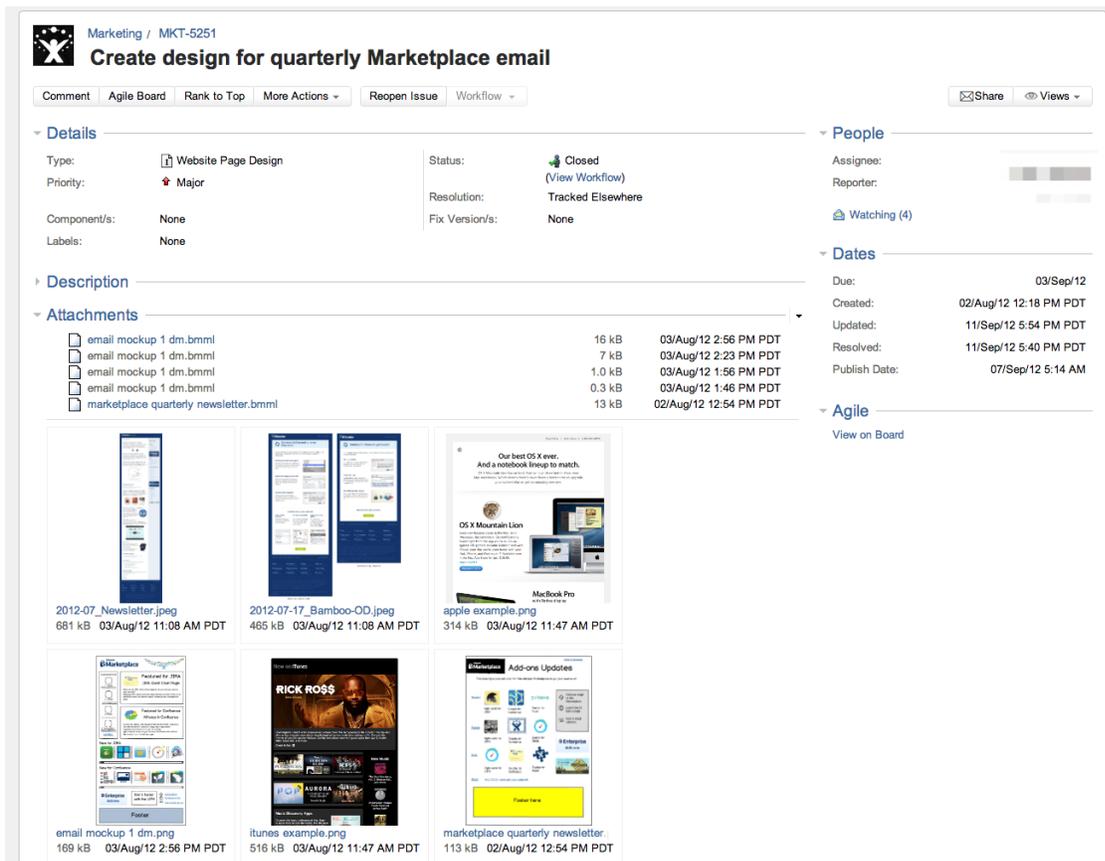
## ADOPT SPRINTS FOR MARKETING TASKS

Although JIRA is often used for tracking issues and bugs on software development projects, JIRA's tracking, commenting, and collaborative features are just as useful for non-technical teams. A good example of that is a marketing campaign that involves multiple departments. With help from Confluence, a marketing plan can be established with task breakdowns, owners, due dates, statuses, comments and so forth.

For each of the task breakdowns you've added in Confluence, you can add JIRA stories with designated owners, estimates and sub-tasks, which can include attachments for design, mockups, wireframes, etc. Once the marketing sprints are added, you're then able to track the progress of the marketing plan versus the anticipated release date.

The following JIRA dashboard gadgets--Sprint health, Sprint burndown, Issues in Progress and Assigned to gadgets-- are all useful in visually

tracking metrics that measure the progress of your marketing project.



The screenshot shows a JIRA issue page for the project 'Marketing / MKT-5251'. The issue title is 'Create design for quarterly Marketplace email'. The issue is in the 'Closed' status. The priority is 'Major'. The type is 'Website Page Design'. The resolution is 'Tracked Elsewhere'. The fix version is 'None'. The issue is assigned to a user. The description is empty. The attachments list includes: 'email mockup 1 dm.bmm1' (16 kB, 03/Aug/12 2:56 PM PDT), 'email mockup 1 dm.bmm1' (7 kB, 03/Aug/12 2:23 PM PDT), 'email mockup 1 dm.bmm1' (1.0 kB, 03/Aug/12 1:56 PM PDT), 'email mockup 1 dm.bmm1' (0.3 kB, 03/Aug/12 1:46 PM PDT), and 'marketplace quarterly newsletter.bmm1' (13 kB, 02/Aug/12 12:54 PM PDT). The attachments are displayed as a grid of six images: '2012-07\_Newsletter.jpeg' (681 kB, 03/Aug/12 11:08 AM PDT), '2012-07-17\_Bamboo-OD.jpeg' (465 kB, 03/Aug/12 11:08 AM PDT), 'apple example.png' (314 kB, 03/Aug/12 11:47 AM PDT), 'email mockup 1 dm.png' (169 kB, 03/Aug/12 2:56 PM PDT), 'itunes example.png' (516 kB, 03/Aug/12 11:47 AM PDT), and 'marketplace quarterly newsletter' (113 kB, 02/Aug/12 12:54 PM PDT).

Image Source: <https://www.atlassian.com/blog/archives/how-to-execute-a-marketing-campaign-using-jira-confluence>

In much the same way JIRA helps product owners on development projects prioritize tasks on the cross-department epic backlog by incorporating feedback from customers, you can also use JIRA's

collaboration features to develop a marketing campaign that can help shape that customer demand.

## Author Bio.

As VP of Engineering, Hamesh is responsible for the worldwide engineering function of Zephyr and brings over 18 years of engineering, mobile, and networking technology experience to this role. Prior to Zephyr, Hamesh held numerous lead engineering roles at Asurion, InnoPath Software, Sonus Networks, and Cisco Systems. Hamesh holds an MS in Computer Science from Texas A&M University.

Outside of work, Hamesh enjoys golfing, playing tennis, and spending time with his two daughters.



### HAMESH CHAWLA

VP of Engineering

**Zephyr**

[hamesh.chawla@getzephyr.com](mailto:hamesh.chawla@getzephyr.com)



# About Zephyr

Zephyr is a leading provider of quality management solutions, powering intelligent DevTestOps for more than 11,000 global customers across 100 countries. Project teams and enterprises of all sizes use Zephyr's products to enable continuous testing throughout their entire software delivery pipeline to release higher quality software, faster. Zephyr is headquartered in San Jose, CA with offices in King of Prussia, PA, Europe and India. For more information, please visit [www.getzephyr.com](http://www.getzephyr.com).



**Contact Zephyr Today!**

[sales@getzephyr.com](mailto:sales@getzephyr.com)

[www.getzephyr.com](http://www.getzephyr.com)

+1-510-400-8656